Raw data captured with USBLyzer

Hexadecimal data							Range (deg)
81	0B	00	FF	FF	A4	FF	899
81	0B	00	FF	FF	86	FF	898
81	0B	00	FF	FF	68	FF	897
81	0B	00	FF	FF	4A	FF	896
81	0B	00	FF	FF	2C	FF	895
81	0B	00	FF	FF	0E	FF	894
81	0B	01	FE	FF	E0	FF	893
81	0B	01	FE	FF	A4	FF	892
81	0B	01	FE	FF	86	FF	891
81	0B	01	FE	FF	68	FF	890
81	0B	01	FE	FF	4A	FF	889
81	0B	01	FE	FF	2C	FF	888
81	0B	01	FE	FF	0E	FF	887
81	0B	02	FD	FF	E0	FF	886
81	0B	02	FD	FF	A4	FF	885

It is clear that sum of 3rd and 4th byte is always FF (255). Range commands are divided to segments by 7 degrees, each new segment increments value of 3rd byte by 1. Segment 0 defines ranges from 899 to 894 degrees, segment 1 from 893 to 887 etc. To set 900 degrees range command 83 00 00 00 00 00 00 00 appears to be used. The 6th byte determines what range exactly to set from the 7 degrees wide segment, only follwing values seem to be valid.

Hex	Dec
E0	224
A4	164
86	134
68	104
4A	74
2C	44
0E	14

How to construct a command to set range

Assuming that the information above is correct, complete and valid for every value the range can have, a command can be constructed by following these steps. Let's try it for 720 degrees.

1. Which segment to use?
   $900 - 720 = 180$

$180 \div 7 = 25.714... \Rightarrow 25$
Use segment number 25.

2. Use the segment number to calculate values for 3^{rd} and 4^{th} byte.
   $255 - 25 = 230$
   $230 = 0xE6$
   $0xFF - 0xE6 = 0x19$
   Set 3^{rd} byte to 0x19 and 4^{th} byte to 0xE6.

3. Set value of the 6^{th} byte.
   $899 - 25 \cdot 7 = 724$
   We want 720 degrees, so we need to set the 6^{th} byte to 0x4A. 0xE0 vaule sets 724 degrees, 0xA4 723 and so on...

Following logic yields command 81 0B 19 E6 FF 4A FF. I verified this particular result and it seems to be correct.

I suppose that Logitech drivers use some niftier algorithm to generate these commands, perhaps converting the range to hexadecimal numbers or everyting to a binary representation would help to figure it out.